AFDELING DER TECHNISCHE NATUURKUNDE

LEX - 11

- Introduction
- Intermediate Guide (IMG)
- Technical Guide (TEG)
- Reference list of Functions

INTRODUCTION 1

Visual Editing. 1
The indexed System Memory File. 2
Programming - A means of automating the facilities of LEX-11. 2
A means of integrating word and data processing 2
Distributed Processing. 3
Summary 3

SYSTEM FEATURES LIST 4

Type of computer. 4
Video Screen. 4
Wide screen mode. 4
Menu Driven 4
Screen editing. 4
Text input. 4
Custom and single keystroke functions 4
Cut and Paste 5
Scroll. 5
Search. 5
Word Wrap 6
Rulers. 6
Automatic titling and pagination. 6
Memory File records 6
Keystores 6
Record Listings 6
Mass Mailing. 6
Forms 6
Selection 7
Calculating Facilities. 7
Basic Graphics. 7
Document and Program storage. 7

THE PARTS OF A WORD PROCESSOR 8

The Computer. 8
The Disk. 8
Safety copies of documents. 9
Document directories and privacy. 9
Document files. 9
Memory File 9
Video Screen. 9
Keyboard. 11
Printer 11

BEGINNERS GUIDE 12

Introduction. 12
General Operation 13
Starting up LEX-11. 13
Menus 13
Using the Keyboard. 13
Moving the Cursor 13
An overview of Document & Letter creation 15
Document Names. 15
Introduction to Rulers. 16
The EXIT Function 17
Creating a Document 18
Text editing functions. 19
Changing an existing Document or Letter 20

Rulers. 20
Ruler Control 23
Functions available for Ruler Control 24
Insert Text mode. 24
Cut and Paste 25
Moving a Column 25
Moving a Paragraph. 26
Cut and Paste functions 26
Merging from one Document to another using Cut and Paste. . . 26
Merging from one Document to another using the Merge function 27
Printing Documents and Letters 28
Number of Printers. 28
Types of Printers 28
Print Time questions. 28
Controlling document printing - pages, titles and merging . . 29
Control Lines 30
Automatic Paragraph numbering 32
Edit Time Pagination. 33
Document formats - storing details. 34
Keystores 35
Setting Keystores 36
Reading Keystores 36
'FIND' Keystore 36
Alternative find giving context 36
Standard Paragraphs 36
Creating Standard Paragraphs. 37
Modifying Standard Paragraphs 37

INTERMEDIATE GUIDE 38

The Calculator 38
Using the Calculator. 38
Name and Address recording system 40
Entering Names and Addresses. 40
Changing Names and Addresses. 41
Viewing a list of Name and Address records. 41
Printing a list of Name and Address records 41
Selection options 41
All records 41
Individual records. 42
Mass Mailing. 42
One-off letters using existing name and address data records. 42
Simple programming of LEX-11. 42
Text and Simple Symbolic Representations in Keystores 43
Creating. 43
Updating. 43
Recalling 44
Plain text in Memory File records 44
Creating a Memory File record 44
Updating a Memory File record 44
Recalling text from the memory file 44
Creating memory file records without asserted line ends . . . 45
Using a Memory File 45

TECHNICAL GUIDE 46

Hard and Soft Interfaces 46
Symbolic representation of LEX-11 functions 47
Programming Levels. 48
Examples of Single Level programs 48
Example of Nested Level programs. 50
Menu modification and creation 51
The Main Menu 51
Table of Menu Function Characters 52
Altering a Menu 52
Creating a new Menu 52
Ascertaining Memory Record Names 52
Setting up Mass Mailing functions 53
Table of record identifier characters (record types). 54
Name and Address records. 54
The Dictionary. 55
Mass Mailing. 56
Recalling parts of Memory File records. 60
Standard forms. 60
Example of Standard Document. 60
List Processing Reports 61
Column totalling in the list processor. 62
Variable length fields. 62
Multi-line reports. 62
Name and Address record listing in full 63
Variable in the Heading and Literals in the formatting lines. 63
Automatic Addressing of a letter 63
System File Menu 65
View System Records 66
Print System Records. 66
Creating a new Memory File. 66
Copy records from one Memory File to another. 67
Change Memory file within LEX-11. 68
Setting up details for different terminal types (VDUs). . . . 68
VDUFORM Part 1. 69
VDUFORM Part 2. 69
VDUFORM Part 3. 71
Internal LEX-11 operation codes 72
VDUFORM Part 4. 73
The **TEXT system record. 74
Installation Notes. 75
RSTS 75
RT-11 and TSX+. 76
RSX11M. 77
VAX-VMS 78

REFERENCE LIST OF FUNCTIONS 80

INTRODUCTION

LEX-11 is a word-processing software system with a wide range of facilities. For editing text it uses video terminals driven by DEC LSI-11, PDP-11 or VAX Computers running under the operating systems RT11, TSX-PLUS, RSTS, RSX, UNIX, or VAX/VMS. The terminals may be of any type so long as they have up and down, left and right cursor control. LEX-11 may be used with a variety of printers.

This adaptability to computers of various sizes, operating systems and peripherals was designed to enable the large number of existing DEC computers to add word processing facilities, at a small incremental cost.

It is also highly competitive with stand alone word processing systems. The advent of small multi-terminal 11/23 computers running RSX or TSX-PLUS increases the usefulness of LEX-11 in the dedicated word processing marketplace.

Documents edited by LEX-11 are held as standard operating system ASCII files and therefore are compatible with standard DEC communication and systems software. LEX-11 requires its indexed Memory File to be on-line in order to hold names and addresses, standard paragraphs, menus, programs and other information required to support its wide range of facilities.

LEX-11 has a number of well developed features such as its calculator, window cut and paste and programmable keys. It also offers some strategic advantages over its competitors mentioned below:

VISUAL EDITING

LEX-11 is a visual editor - "what you see is what you get."

Except for the brief appearance of control information, or messages, the entire information on the screen is an exact window on the document being created or changed. Unlike some other word processing systems, there is almost a complete absence of command codes embedded in the text.

Justification takes place in front of your eyes. You view the actual layout as you create or edit text. The only characters which need manipulation are visible on the screen. 'Invisible characters' such as carriage return, line feed or tab simply do not exist in the text. Nevertheless they can be used as required to move the cursor (the flashing line or rectangle indicating the typing position) to suitable points on a line but they are not embedded in the text. Blank portions of text consist solely of spaces. This removes a common difficulty in editing - of stumbling on 'invisible' characters.

LEX-11 is truly two dimensional.

An editing operation can start anywhere on the screen and is not limited to the bottom line. This makes editing both faster and easier on your eyes. You can progress not only horizontally but vertically or in a combination of both. LEX-11 is, therefore, suitable for elementary graphics, e.g., lines, bar charts, flow diagrams or replicating pictorial figures.

The document text may be scrolled forwards to the end of the file. Although the text may in general be scrolled backwards by only three whole screens, it has the advantage that a standard operating system file structure is used which maintains compatibility with standard DEC software including communications routines. Also file size is virtually unlimited. Automatic return to beginning of document is provided and then search mode can be used to position the edit window at a desired position in the document.

THE INDEXED SYSTEM MEMORY FILE

Records on LEX-11's Memory File are indexed and, therefore, may be immediately accessed. Hence, user record-keeping is simple, e.g., personnel records may be recalled via their key, and changed in random sequence with virtually immediate response.

System records for insertion or modification are displayed on the screen clearly formatted by forms designed and implemented by the user. Since the user may also design his own simple reports, small complete data-processing systems may be constructed. One Memory File can store record types and layouts of many different kinds. Therefore, it is possible, for instance, to implement an invoicing application which directly accesses any given name and address, via its access key. LEX-11 can also directly access any number of line items via a product code displaying the description and price. Moreover, after the user has input the quantity, it can perform the extension and subsequent totalling of the invoice.

In most word processing systems, name and address records are stored serially and it can take 5 to 20 minutes to locate any particular record which inhibits their use for on-line record-keeping applications.

One imaginative use of LEX-11 in record-keeping was implemented by an organization in London, England, which keeps details on some hundreds of small companies. Some of these companies may have up to 50 directors or shareholders. LEX-11 can immediately access the information held on any particular company via either an abbreviation of the company name or its number, and display the main data on a single screen. A small menu at the top of this screen allows, at a single key-stroke, the subsequent display of directors or shareholders names and addresses, up to 8 at a time. Further, relevant names and addresses are called up again at the touch of a key. This type of elaborate record-keeping is only possible with an indexed memory file, combined with key-stroke programming, cut and paste and an on-screen calculator.

PROGRAMMING - A MEANS OF AUTOMATING THE FACILITIES OF LEX-11.

As implied by the above examples, LEX-11 can be programmed to carry out simple or complex tasks automatically. The "programming" consists simply of storing the function key strokes which would otherwise have been used by the typist.

Programs can be stored in and recalled from "key-stores" or the Memory File. There are ten key-stores which can hold up to 78 key-strokes each so that they can also be used to hold commonly recurring words and phrases. These key-stores are held in the central memory of the computer, and so reading from them is very fast. A common technique is to store all such programs on the Memory File and then to load a particular program into the key-stores when it is to be executed.

A MEANS OF INTEGRATING WORD AND DATA PROCESSING

LEX-11 is written in a decision table language which allows very large programs, such as LEX-11, to be run even on an LSI-11, because it employs paging. Since LEX-11 under RT11, RSTS, RSX and VMS uses standard operating system files for its documents, it can be integrated with existing communications software such as DECNET.

LEX-11's report generator can produce reports from selected name and address or other data records held on a Memory File. Sorting and sub-totalling will become available in LEX-11 in the near future.

DISTRIBUTED PROCESSING

This close integration of word processing and data processing makes a micro computer with LEX-11 an effective tool for distributed processing. A number of different sized computers ranging from the LSI-11 and 11/23 to the 11/44 and VAX computers can all use LEX-11, and other programs written in other languages, and will be able to be linked with DECNET. Growth of any particular node can thus be accomplished with minimal disruption because of this large range of compatible systems.

SOFT INTERFACES

Since LEX-11 does not depend on any particular video terminal, its use with existing hardware is much simplified. New videos may also be added easily because their interface to LEX-11 is soft. This flexibility is provided by the user specifying on a formatted screen the various characteristics of his particular video. Different videos, and identical videos in different modes, can run off one Memory File if desired.

This same 'soft-interface' idea is applied to the system messages of LEX-11, allowing it to be adapted to user preferences, or to languages other than English.

The menu structure of LEX-11 is also 'soft' and, hence, this important man-machine interface can once again be adapted by the user to his particular requirements. Internal sub-systems such as invoicing and external sub-systems using the link to other programs may be incorporated in LEX-11's menus by the user.

SUMMARY

LEX-11 may be used as a stand alone word processor or as an add-on facility to existing PDP-11 or VAX computers. A variety of direct access record-keeping systems may be easily developed in LEX-11. It is also suitable for applications which require screen processing, direct access to more than one record type, a calculator and programming. An example is invoicing where name, address, and product records must be accessed and calculations performed.

Word and data processing may be integrated because LEX-11 uses standard operating system files which can be integrated with communications. LEX-11 can run simultaneously, or alternately on small computers, with other commercial data-processing programs, including data-entry. Hence, LEX-11 is well suited for use in distributed processing systems.

pagina's 4 t/m 37

vormen "origineel" tbv.

"Gedeelte uit de LEX-u Uyen's Sprizle"

THE CALCULATOR

The calculator behaves like a pocket calculator, displaying a single number in an accumulator - which is identified on the screen by the cursor situated on the least significant digit (LSD).

The calculator may be used for totalling rows and columns of figures displayed on the screen and for miscellaneous adding, subtracting, multiplying and dividing. Entering and exiting from the calculator are achieved by two specific commands. The ability to enter, perform one or more arithmetic functions, move the cursor by keyboard entries or under the control of a command string in a keystore (see Technical Guide), provides the means to largely automate the production of invoices and other operations.

USING THE CALCULATOR

The calculator may be used by two different methods - typing ESC followed by + (which allows calculations to be performed 'on the spot' - that is, a tab, space or carriage return will exit calculator mode), or by putting hashes in the ruler line, thereby creating columns in which only numbers can be typed.

To enter the calculator without using a calculator-associated ruler, press ESC followed by +. Any figure entered will automatically default to two decimal places, unless you specify Pn (n = number of decimal places - up to seven). A zero appears at the cursor position and any numeric string typed is pushed to the left of the cursor, hence the cursor should have been positioned several places in from the lefthand margin before entry to the calculator. This display of a number represents the display of a pocket calculator and can be cleared by pressing C. The operators such as + - * and / or brackets are not displayed, as is the case with a pocket calculator. The following tables illustrate how to enter the calculator and perform calculations (a) and (b):

(a) 100, plus 50 divided by 5
i.e. 100+(50/5)

KEY	DISPLAY
	| <-----cursor position
ESC+	0
100	100
+	100.00
50	50
/	50.00
5	5
=	110.00
(sp)	110.00 (press space bar to exit calculator mode)

(b) 100 plus 50, divided by 5
i.e. (100+50)/5

KEY	DISPLAY
	|
ESC+	0
(	0
100	100
+	100.00
50	50
)	150.00
/	150.00
5	5
=	30.00
(sp)	30.00

Up to four sets of nested round brackets may be used, e.g. 3(4+R1(R2/7))

USING THE CALCULATOR UNDER A RULER CONTAINING 'HASHES' (see Section 3.4)

Set up a ruler appropriate to the job you want to carry out. The hash columns in the ruler must be separated by at least two '.'s.

J.....M.....#####.##.....#####.##.....########.##.....			
Tab to first set of hashes - this will enter calculator mode	12.87 (type no.)	3.69 (type no.)	16.56 (press H)
Repeat entry of numbers	6.11	837.44	843.55
To total vertically, tab to each column and press V	18.98	841.13	860.11

TABLE OF CALCULATOR FUNCTIONS

Keyboard	Action
ESC+	Enter calculator mode
tab, space or c/r	Exit calculator mode
+	Add
-	Subtract
*	Multiply
/	Divide
=	Display current result
(	Open of bracketed expression
)	Close of bracketed expression
C	Clears the accumulator
%	Percent, e.g. 500*50%=250
V	Calculates and displays vertical total - that is, of all numbers in column above cursor (which should be positioned under the righthand digit). Any non-numeric string will determine the top of the column
H	Calculates and displays horizontal total - that is, of all numbers in the same row to left of cursor. Any non-numeric string will terminate totalling
N	Negate (Reverse Sign) Same as +/- key on a calculator
Pn	Sets n decimal places (default is 2 places)
An	Add number into store n
Sn	Store number in store n
Rn	Recall from store n to accumulator

NAME AND ADDRESS RECORDING SYSTEM

Names and addresses used frequently can be recorded and recalled for one-off letters, mass mailing or report listings. For the latter two cases, a powerful selection criteria is provided.

The Mass Mailing Menu is shown below:

LEX-11 Mass Mailing System Menu

N - Name & Address file create/update
C - Create a new letter to existing contact
L - Print letters to selected addresses
S - Name & Address file selective print
Q - Name & Address file selective view
1 - Amend output formats

. - Select required option

ENTERING NAMES AND ADDRESSES

Option N in the Mass Mailing Menu brings the following format onto the screen for entry of names and addresses and its use is much the same as a card index:

```
NAME KEY  :[ _________
NAME      : ______________________________
ADDRESS   : ______________________________
            ______________________________
            ______________________________
            ______________________________
                                  PHONE: ___________
                                  TELEX: _______

CONTACT 1): ______________________________ ______________________________
            ______________________________

CONTACT 2): ______________________________ ______________________________
            ______________________________

CONTACT 3): ______________________________ ______________________________
            ______________________________

INSERT  1): ____________________________________________________________
INSERT  2): ____________________________________________________________
INSERT  3): ____________________________________________________________

                                  Write Record _
```

The cursor will have been positioned on the key. Enter a key of up to nine characters - this can be anything you choose, using letters or numbers, to identify the record. Lists produced using the Name and Address records will be in key order, therefore the key chosen should be in the required alphabetic or numeric sequence.

If the key is nine characters long, the cursor will automatically move to the next line, otherwise use a carriage return. Enter all your details, using carriage returns to move to fields on the next line or tabs to move to a separate field on the same line. The Insert 1), 2) and 3) lines are for entry of information that is particular to that record which you may want to include in a mass mailing letter.

When you have entered all the details you have, position the cursor on the 'Write Record _' mark by using CTRL L, then press ESC]C (for create). The details will disappear from the screen as the cursor moves through the record and the cursor then moves back to the key ready for entry of the next record. If the key already exists, the terminal will bleep and the details will be left on the screen, i.e. they will not be written away, so that you can decide whether you want to update the existing record or alter the key.

When you have finished entering data, press ESC E to return to the menu.

CHANGING NAMES AND ADDRESSES

Alterations, additions or deletions can be made to name and address records by again selecting Option N (name and address file create/modify).

Enter the key of the record, then press ESC]R. The record will appear on the screen with the cursor positioned on 'Write record_'. Move the cursor around the screen with the arrow keys, making the changes you require, then use CTRL L to position the cursor on 'Write record_' and press ESC]U.

Again, use ESC E to return to the menu when you have finished.

VIEWING A LIST OF NAME AND ADDRESS RECORDS

A list of the names and addresses recorded can be viewed by choosing Option Q in the Mass Mailing Menu. (If the format currently available for listing the details is not as you would want it, this can be changed - see Technical Guide). A choice of all names and addresses or selected ones by fields contained within them can be made - the use of this selection option is described below.

PRINTING A LIST OF NAME AND ADDRESS RECORDS

The S option in the mass mailing menu prints out a list of the Names and Addresses recorded by whatever selection criteria you choose. (Again, the format for the details to be printed can be changed to whatever you require - see Technical Guide).

SELECTION OPTIONS

ALL RECORDS

After selecting Option Q or S, then in response to the question 'Select.....' a carriage return instructs LEX-11 to list all the records stored. All records starting with, say, B can be listed by typing in capitals (after Select) KEY>B AND KEY<C.

INDIVIDUAL RECORDS

Individually selected records can be listed if, in response to 'Select.....', you answer with the customer keys you require. For instance, if you answer with KEY=SMITH,JONES then only those two records will be listed.

It is also possible to select records to be listed by information contained within them, such as a country, by answering 'Select.....' by typing ADDRESS:Germany AND NAME:Smith (address contains Germany and name contains Smith). The selection criteria are = : < > and may be linked with AND/OR/NOT/AND NOT. The expression CONTACT2 NOT=vvv (where v represents depressions of the space bar) will cause only records which have something entered in CONTACT2 to be listed.

MASS MAILING

Option L in the Mass Mailing Menu is used for mass mailings. It allows linking of an existing letter with name and address records for mass mailing.

(The letter should have first have been created through the Create option in the main menu - just the body of the letter without any addressee details because they will be picked up from the Name and Address record - and if any variable information entered in Insert 1), 2) or 3) should be included this should be embedded in the text by typing <INSERT1>, <INSERT2> or <INSERT3> or a combination of any of them).

After selecting L, you will be asked to give the name of the letter you wish to mass mail, following which 'Select.....' will appear. Answer in the same way as before. The names and addresses selected will be printed according to the format set up for option L (again these details can be changed - see Technical Section).

ONE-OFF LETTERS USING EXISTING NAME AND ADDRESS DATA RECORDS

Option C in the Mass Mailing Menu allows the creation of a letter to a recipient whose details are already stored. It is different from a mass mailing in that it merely recalls the name and address details onto the top of a letter being created on the screen to save the time and effort of typing these details in again when they already exist.

Press Option C in the Mass Mailing Menu (it is exactly the same as using create, option C, in the main menu except that it ties in with the name and address records). Enter a new document name for the letter required. You are then asked for the key of the name and address record you wish to access. Enter this, then press carriage return. Immediately, the name, address, date and Dear .... will be inserted at the top of your screen. When you have finished, exit with ESC E and you have a letter stored away for printing out or re-editing in the same way as if you had used the normal create option.

SIMPLE PROGRAMMING OF LEX-11

By programming LEX-11, we mean creating and storing strings of characters representing keyboard key strokes, temporarily within the keystores or

permanently in the Memory File. These strings contain symbolic representations of LEX-11 functions as well as ordinary plain text, and can be recalled later for execution and/or display. Their complexity can vary widely. The simpler of them are just strings of characters stored away to be re-displayed later, here referred to as plain text. The complexity of such programs is unlimited. Some of the most complex programs include nested calls to other programs, other program creation, automatic on-screen calculations, index key Memory File access/update etc. (For the moment there are no decision functions in this key-stroke mode of programming).

TEXT AND SIMPLE SYMBOLIC REPRESENTATIONS IN KEYSTORES

CREATING

Frequently-used text of up to 78 characters can be stored away within the keystores. The storing in one of the 10 keystores is done by typing:

ESCSn ...text... (carriage return)

(where n (0 - 9) is the number of the selected keystore.

The symbolic representation of carriage return is \, therefore letter salutations can be stored within keystores, e.g.

ESCSn Yours truly,\\\\\J. Doe,\President (carriage return)

A single line of text which is already on the screen can be stored by combining cut and paste and a keystore. Do this by moving the cursor over the first character and pressing ESC (, then moving the cursor to the end of the text and pressing ESC)nL which will duplicate that text directly into the selected keystore:

ESC (...line-of-text... ESC)nL

(n is the number of the keystore)

UPDATING

The contents of the keystores can be edited by typing:

ESCSn

which will cause

n=.....old text.....

or n=Yours truly,\\\\\J. Doe,\President

to be displayed on the current line at the extreme left of the screen over the top of any existing text (the existing text is not overwritten, the keystore text is displayed temporarily on top of it). The user can then use the arrow keys to position the cursor and overtype parts or all of the old text. The edited text is stored back by pressing return.

RECALLING

The text from the keystore is recalled by positioning the cursor where you want the text to be displayed and typing:

ESCn

The displayed text will be subject to any justification implied by the current ruler and to the current mode setting (insert or overtype).

e.g. Yours truly,

J. Doe,
President

PLAIN TEXT IN MEMORY FILE RECORDS

CREATING A MEMORY FILE RECORD

Text of more than 78 characters or any multi-line text to be kept and used frequently must be stored in the Memory File, as opposed to in keystores. The size is limited to one screenful of text. The following is typed to create a new Memory File record.

```
[KEY.......
text etc. .......  .........  ..........  .........  .....
........  .......  .........  ..........  .........  .....
........  .......  .........  ..........  .........
ESC]C
```

The text between the [and ESC]C will be stored away. If the text already exists on the screen, one simply types [KEY at the beginning and ESC]U at the end. Note that a carriage return will be stored at the end of each line.

UPDATING A MEMORY FILE RECORD

An already existing Memory File record is brought back on the screen by typing:

[KEY ESC]R

The contents of the record are then displayed on the screen for editing. After editing is finished, the old record is overwritten by positioning the cursor on the line following the last line of text and pressing ESC]U. (Until that command is executed, the changes have occurred on the screen only and will not be stored if LEX is exited).

RECALLING TEXT FROM THE MEMORY FILE (for example, into a document)

The Memory File record is recalled on the screen by typing:

[KEY]

The text will be recalled subject to the current mode (insert or overtype) setting; however, note that the original line-ends (carriage returns) will still be present in the text and will be asserted.

CREATING AND UPDATING MEMORY FILE RECORDS WITHOUT ASSERTED LINE-ENDS

The Memory File Records can be stored away without the carriage return by using ESC>C and ESC>U respectively (instead of using a square bracket). Text recalled onto the screen from records created or updated in this way then conforms to the ruler in use at the time of recall. This is a necessary way of creating Standard Paragraphs, since they will probably be recalled into different documents with a variety of rulers in use within them.

USING A MEMORY FILE

A Memory File is a keyed direct access file (called LEX11.VMF or some other name with a .VMF file extension) used to store variable length data records of up to a screenful of characters. The user may have any number of Memory Files but only one can be used at a time from a particular terminal.

Every record stored on a Memory File has associated with it a 10 character key which is used to label the record as it is being stored or recalled.

When storing a record in the Memory File, strings of spaces in the record are squashed to a single character before writing the record onto a disk. Therefore, it is largely the number of non-space characters which will determine the file size.

Apart from the records created by the users, the Memory File contains system records describing the word processing input/output and working environments to LEX-11. All the Menus and Keyboard hardware descriptions, for example, are kept in the Memory File and the user can easily change them using the Memory File access mechanisms described in the following paragraphs.

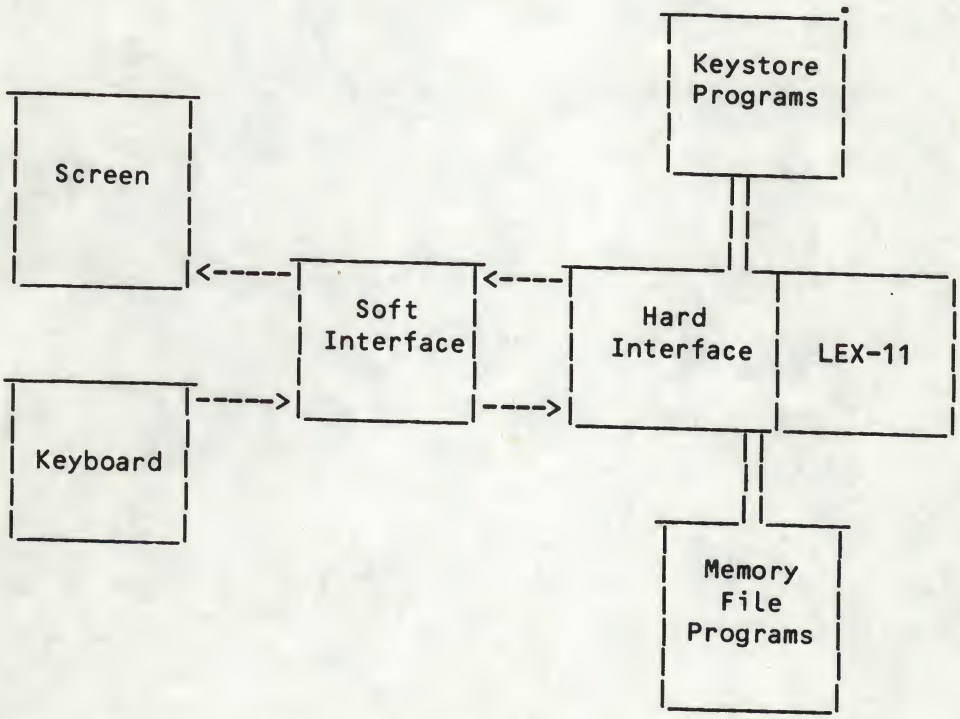
HARD AND SOFT INTERFACES

To enter multi-key word processing functions directly from the terminal, special keys can be used (i.e. Auxiliary keypad and arrow keys which generate 'escape' sequences automatically).

The special keys transmit ASCII 'control' characters to LEX-11. Some of them, e.g. ESC, or CTRL pressed simultaneously with another key) transmit a single 'control' character to LEX-11, others a sequence consisting of one special character (usually ESC) followed by other character(s). These ASCII 'control' characters or sequences are not displayed on the screen but are directly interpreted by LEX-11. In fact, they cannot be displayed since they have no visible representation.

LEX-11 is delivered with one 'hard' internal code per function firmly set within its 'hard' interface program code. Some of the 'hard' internal codes are single ASCII 'control' characters, others are short 'escape' sequences. The Memory File delivered with LEX-11 contains one or more sets of 'soft' keyboard interfaces which can be redefined or added to by advanced users via the Soft Interface facility called from the LEX-11 System Menu.

The 'soft' interfaces are usually used to introduce to LEX-11 the special keys that generate escape sequences automatically (e.g. the auxiliary keypad and arrow keys of the VT100) or to re-allocate the LEX-11 functions onto different locations on the main keyboard. There are more functions in LEX-11 than special keys available and, therefore, it is necessary to generate some of the escape sequences manually by pressing ESC followed by one or more main keyboard keys. (When using keyboards without any auxiliary keypads, it is necessary to generate the ASCII 'control' characters directly from the main keyboard). The following diagram represents the flow of data within the LEX-11 system:



Functional Diagram of LEX-11 Interfaces

As shown in the diagram above, the data coming from the keystore or Memory File does not pass through the 'soft' definable interface and, therefore, all

LEX-11 programming codes (described in the following paragraphs) stored within them have to be based on 'hard' interface codes. Thus a stable programming environment not influenced in any way by changes in the 'soft' interface takes precedence over the 'hard' one.

All data coming from the keyboard passes through the 'soft' interface. The 'soft' defined escape sequences are transformed by it into 'hard' internal codes and passed on for execution. As far as the keyboard user is concerned, the 'soft' interface takes precedence over the 'hard' one.

SYMBOLIC REPRESENTATION OF LEX-11 FUNCTIONS

To be able to construct (i.e. to enter, edit and recall) and execute a LEX-11 program, one represents the functions symbolically. This symbolic code prevents immediate interpretation of the functions while they are being typed-in. In addition, the symbolic code is displayable on the screen and can be visually inspected and edited.

What we call the program is nothing more than stored text and function symbols enclosed by " at the beginning and " at the end. Such a program when recalled from a keystore by ESCn or from Memory File by [KEY] will drive LEX-11 directly as if the characters and function calls were arriving from the keyboard at a very high speed.

Both the keystores and Memory File can hold the LEX-11 programs within them. To create, update and recall programs (i.e. to create, update and recall keystores and Memory File records with embedded LEX-11 functions) basically the same steps are taken as the ones described earlier in the paragraphs on 'Plain Text'. However, one has to give special attention to correct embedding of the function calls and the right choice between ESC]R and], all of which is explained in the following paragraphs.

Symbol	Represents	LEX-11 Action
'	Followed by a capital letter A to Z represents an ASCII control character, e.g. 'A stands for Ctrl A, 'T for Ctrl T, etc.	Execute the corresponding Control function, e.g. 'A recalls function Insert Line, 'T recalls function Top of Screen
$	ESCAPE character. Must be followed by one or more characters of the escape sequence. (Since the ESC is an ASCII Ctrl[, it could be represented by '[- however the $ is used for short). e.g. $+ stands for ESC+, $1 stands for ESC1, $UB stands for ESC UB etc.	Execute the corresponding ESCAPE function, e.g. $+ enters the calculator $1 recalls keystore 1 $UB recalls unset background, etc.
!	Represents ?	Halt the program execution and prompt the keyboard operator for data, then resume the program when the return key is pressed.

<	Represents [	Execute [(memory record start)
>	Represents]	Execute] (memory record read)
}	Is short for 'H	Execute left arrow
\	Is short for 'M	Execute carriage return
|	Is short for 'I	Execute tab

PROGRAMMING LEVELS

At the commencement of a keystore or memory read, LEX-11 will interpret these special characters as a LEX-11 symbolic function code, until a " (double quote) is encountered, after which LEX-11 will interpret them as actual characters. A further occurance of the " will reset them to functions. The character @ , when in function mode will cause the next character to be interpreted as a character, thus allowing an additional level of nesting.

CAUTION: Only 'hard' internal codes are accessible from the program. The 'soft' internal codes are terminal-dependent and as such are unsuitable for programming. For example, the calculator can be entered from the keyboard by typing ESC= as well as by typing ESC+ but from the program it is accessible by $+ only.

The nested programming levels are essential for writing programs which are to be stored on the Memory File but when executed actually create other programs in keystores or other Memory File records.

EXAMPLES OF SINGLE LEVEL PROGRAMS

1) The typing of a letterhead can be improved by inserting programming codes to ask for data to be inserted.

```
[Ltrhead

        D O U G H N U T  W H O L E S A L E R S  L T D.
              (A Division of D.G.& Z. Ltd.)
        ------------------------------------------------

                                        Please address all
                                        correspondence to:

                                             P.O. Box 12,
                                             London Road,
                                                Newtown.

                                        Telephone 140642
Dear !,
                                   Date  !
ESC]U
```

When this letterhead is recalled by [Ltrhead], the text up to 'Dear' is displayed and then execution is stopped by the question mark which allows the user to type in data followed by carriage return. Having entered the name terminated by carriage return,the text up to 'Date' is displayed and another pause occurs. After typing in the date and carriage return, the cursor is positioned ready for the rest of the letter to be typed in.

2) The following version of the same example demonstrates automatic retrieval of clients' addresses from the Memory File:

```
[Ltrhead

                D O U G H N U T   W H O L E S A L E R S   L T D.
                      (A Division of D.G.& Z. Ltd.)
                ------------------------------------------------

                                          Please address all
Addressed to:                             correspondence to:

<!$>R                                          P.O. Box 12,
                                               London Road,
                                                   Newtown.

                                            Telephone 140642

Dear !,
                                          Date !

ESC]U
```

When the 'Ltrhead' is recalled by [Ltrhead], the first pause occurs in <!$>R. (This string of symbolic programming codes stands for [!ESC]R). Having requested and obtained the client's key (e.g. DOE) from the keyboard, LEX-11 retrieves the contents of the record and places them in the lefthand corner of the letter overwriting the client's key just typed in. The rest of the letterhead is identical to the previous example.

The client address records would typically be created as follows:

```
[ Doe
Mr. John Doe,
Doe & Doe Ltd.
P.O. Box 54,
School Lane,
Newtown.
ESC]C
```

3) To enter a price and a quantity and automatically extend the product, keystore 1 could be set as follows:

```
ESCS1A=||$+!S1 |B=||$+!S2 |A*B=||$+R1*R2=  (carriage return)
```

The actions performed by this program are as follows:

a. Display the words "A="
b. || will make the cursor skip two tabs
c. $+ enters the calculator
d. The ! is interpreted as a pause point for data entry by the operator. The operator types in a number then presses carriage return to recommence recall of the stored program
e. The S1 will cause the calculator to store the number the cursor is pointing at in Register no.1 and the following space will step out of calculator mode
f. | will make the cursor tab once, then after displaying B=, || will make the cursor skip two more tabs and the calculator is entered. The program prompts the operator for the second number, which when entered and carriage return is pressed, will be stored in Register no.2. The following space steps out of calculator mode, then another tab positions the cursor and A*B= is displayed
g. When the calculator is entered for the third time, the R1*R2= will recall and multiply the contents of the Registers 1 and 2 and display the result on the screen. After that, the final space will step out of the calculator mode and the programming mode is then terminated

EXAMPLE OF NESTED LEVEL PROGRAMS

This example of a nested level program is entered into the Memory File under the key 'EXAMPLE'. When EXAMPLE is recalled, it will store programs in keystores 1 and 2 which in turn perform calculations similar to the previous example.

```
[EXAMPLE
$S1"$U7\Enter\A=|||!{$+S1\B=|||!{$+S2\\A*B=||$+R1*R2=\"\
Keystore 1 set.\$S2"$U7\Enter\A=|||!{$+S1\B=|||!{$+S2\\
A*B=||$+R1*R2=\"\Keystore 2 set.\
ESC]C
```

When recalled by: [EXAMPLE] the Memory File program will set keystores 1 and 2 by executing the following actions:

```
ESCS1$U7\Enter\A=|||!{$+S1\B=|||!{$+S2\\A*B=||$+R1*R2=\<return>
Keystore 1 set.<return>
ESCS2$U7\Enter\A=|||!{$+S1\B=|||!{$+S2\\A*B=||$+R1*R2=\<return>
Keystore 2 set.<return>
```

After which, the screen will display:

```
Keystore 1 set.
Keystore 2 set.
```

When invoked by ESC1, the program will execute functions:

```
ESCU7            i.e. use all-tabs ruler
Enter (carriage return)
A=(tab)(tab)(pause point)(backspace)(enter calculator)S1(return)
B=(tab)(tab)(pause point)(backspace)(enter calculator)S2(return)
(return)
A*B=(tab)(tab)(enter calculator)R1*R2=(return)
```

After which the screen will typically display:

```
Enter                          .... executed by ESC1
A=          123.5
B=          2345.67

A*B=        289690.24


Enter                          .... executed by ESC2
A=          34.78
B=          123456.78

A*B=        123491.56
```

MENU MODIFICATION AND CREATION

THE MAIN MENU

Unlike other menus, the Memory File key of the Main Menu is fixed - it is always called **MENU. It may be displayed for editing in the same way as any other Memory File record, i.e. by typing [**MENU ESC]R - this might produce:

```
[**MENU     .
C           C - Create a new document
E           E - Edit an existing document
V           V - View a document
PP          P - Print a document
QP          N - Multiple print

A*COPY      A - Copy a document
A           X - Copy and Edit a document
K           K - Delete a document
R           R - Rename a document
X*PAPER     D - Amend document printing details

S*VDTI      I - View document index (Full)
X*DTIR      Q - Quick document index
M*HELPMENU  Z - Help File Menu
M*MAILSHOT  M - Mass Mailing System
M*SYSTEM    S - System Menu

            . - Select option
```

Notes:

The first 11 columns are not displayed when the menu in actual use.

To write the menu to the Memory File, presumably after making some changes, position the cursor on the far left of the screen on line below the last option, i.e. below the Select option, as for any write to the memory and press ESC]U

The layout of each line, apart from blank lines or lines with blanks in the first 12 columns, consists of a one character function code followed optionally by the up-to-ten character key of a Memory File record, followed mandatorily by an option code. The text following character 12 is purely descriptive and has no effect on the functioning of the menu.
When LEX-11 is requesting a menu selection, it is possible to type '/' followed by any letter that is valid in the first position of a menu line. This switches LEX into the mode of operation selected, and is a useful means of recovering from the situation that can occur if you corrupt **MENU. To get into memory file edit mode in this situation, type /X. To change VDU types you use /U.

TABLE OF MENU FUNCTION CHARACTERS

Function Character	Parameter Following (O = Optional, M = Mandatory)	Action
E	O	Document Edit
C	O	Document Create
P		Document Print
V	O	Document View
A	O	Document Edit, with output to a new file
S	M	Calls the LEX-11 Select function. It must always be followed by the key of a LEX-11 Report Format
X	O	Enter the playground. If a key follows, LEX-11 executes the corresponding record as a program.
M	M	Display the Menu relating to the following key
L	M	Mailmerge using the format specified
R		Document rename
K		Document delete
T		Display Memory File Statistics
G	M	Chain to another program. Followed by program name
F	M	Swap Memory Files. Followed by memory file name
U	M	Change VDU type. Followed by VDU type record name
3		Memory File Create
4		Memory File Copy

ALTERING A MENU

To alter an existing menu, enter option 1 in the memory file, type [KEY ESC]R to read the record onto the screen and make the desired changes using the normal editing method. As described in the notes against the Master Menu above, position the cursor immediately after the select option field and type ESC]U.

CREATING A NEW MENU

To create a new menu, it is suggested that you enter option 1 in the memory file, recall a similar menu and alter the key and the contents as desired and write the new record away to the memory file as described above. (Recalling a memory record onto the screen merely brings back a copy and no changes are recorded in the record until the keys ESC]U have been pressed. Therefore, provided the key is changed immediately to a new one, the copy can be used as the basis of a new memory record and written away with the usual command - ESC]C, thus creating a new record and leaving the original one intact).

ASCERTAINING MEMORY RECORD NAMES

By recalling the Master Menu memory record onto the screen, the names of any memory records used by the Master Menu will also be displayed. If the format or menu name you seek is not in the Master Menu system record, then recall the menu system record shown on the Master Menu which contains the name of the system record you seek. It may of course be necessary to read in more than one menu record, depending on the depth to which menus have been nested.

e.g. to get to the menu system record which allows options to be chosen for viewing lists of names and addresses, recall **MENU:

```
[**MENU      .
C            C - Create a new document
E            E - Edit an existing document
V            V - View a document
PP           P - Print a document
QP           N - Multiple print

A*COPY       A - Copy a document
A            X - Copy and Edit a document
K            K - Delete a document
R            R - Rename a document
X*PAPER      D - Amend document printing details

S*VDTI       I - View document index (Full)
X*DTIR       Q - Quick document index
M*HELPMENU   Z - Help File Menu
M*MAILSHOT   M - Mass Mailing System
M*SYSTEM     S - System Menu
C*GRAPH      G - Create a graph

             . - Select option
```

Then recall *MAILSHOT:

```
[*MAILSHOT LEX-11  Mass Mailing System Menu
X*FORM1      N - Name & Address file create/update
C*LETTR      C - Create a new letter to existing contact
M*INDXL      L - Print letters to selected addresses
M*INDXS      S - Name & Address file selective print
M*INDXQ      Q - Name & Address file selective view
X            1 - Amend output formats

             . - Select required option
```

Next, recall *INDXQ:

```
[*indxq      Name & Address File View
S*OUTQ1      1 - Name & Phone

S*OUTQ2      2 - Name & Contacts

             . - Select Format
```

SETTING UP MASS MAILING FUNCTIONS

The Mass Mailing menu as released is shown below as it appears after recall onto the screen, i.e. after typing [*MAILSHOT ESC]R

```
[*MAILSHOT LEX-11  Mass Mailing System Menu
X*FORM1      N - Name & Address file create/update
C*LETTR      C - Create a new letter to existing contact
M*INDXL      L - Print letters to selected addresses
M*INDXS      S - Name & Address file selective print
M*INDXQ      Q - Name & Address file selective view
X            1 - Amend output formats

             . - Select required option
```

Options N, S, Q and L show facilities for maintaining, displaying and printing Names and Addresses in various formats.

All of the Mass Mailing Forms and Formats which currently exist in the Memory File of LEX-11 can be changed and new ones created to suit the user's own requirements. The current ones are: *FORM1 which is the system record format designed for entry of name and address data, *DICT defines it, *INDXQ is a subsidiary menu that allows selection of *OUTQ1, *OUTQ2 or *OUTQ3 (list viewing options), *INDXS is a subsidiary menu allowing selection of *OUTS1, *OUTS2 or *OUTS3 (list printing options), and *INDXL is a subsidiary menu allowing selection of *OUTL1, *OUTL2 or *OUTL3 (mass mailing options that allow name and address details to be linked with a prestored letter).

e.g.

*FORM1
(*DICT)

*INDXQ *INDXS *INDXL

*OUTQ1 *OUTQ2 *OUTQ3 *OUTS1 *OUTS2 *OUTS3 *OUTL1 *OUTL2 *OUTL3

There is no limit to the number of Mass Mailing records, other than the size of the Memory File and this can be increased according to the amount of space available on disk within the restrictions imposed by the operating system. The records comprising each Mass Mailing or List Processing File must be identified by the first character of the ten character key (the record type), leaving nine characters for the user to define, and this is then included in the heading of the Output Files *OUTQ1, *OUTS1, *OUTL1 etc. so that selection of records automatically finds only those beginning with that record type. For instance, the Name and Address records all start with a blank on the standard Memory File, i.e. NAME KEY :< _____ . A table of these first characters follows:

TABLE OF RECORD IDENTIFIER CHARACTERS (RECORD TYPES)

Character	Description
*	System Records, e.g. Menus, Formats, Programs
*.xxx	Document Class Defaults - (xxx represents the document type , e.g. *.LTR, *.DOC)
:	Document Names (for inclusion in the index of names)
	Name and Address Records
P	Standard Paragraphs
)	Terminal Form Records

The following is a description of the Name and Address Record File, as contained on the standard Memory File, to illustrate to the user how memory record files are built up and their purpose.

NAME AND ADDRESS RECORDS

The system record used to set up the input form for names and addresses, under Option N of the Mass Mailing Menu, is *FORM1. This was set up as below because it suited many of the users of LEX-11 but it can be changed to any form the user requires. It can be viewed or edited in option 1 of the System Menu by typing [*FORM1 ESC]R.

```
[*FORM1    $CB$SBNAME KEY  :< _________
NAME       : ______________________________
ADDRESS    : ______________________________
             ______________________________
             ______________________________
             ______________________________   PHONE: ___________
                                              TELEX: ________

CONTACT 1): ____________________________ ____________________________
            ____________________________

CONTACT 2): ____________________________ ____________________________
            ____________________________

CONTACT 3): ____________________________ ____________________________
            ____________________________

INSERT  1): ____________________________________________________________
INSERT  2): ____________________________________________________________
INSERT  3): ____________________________________________________________

                                         Write Record _'T$SP$UB
```

Changes can be made very easily by merely moving the cursor into the record and changing, adding or removing the relevant fields (the only limit being a screenful). Write the new *FORM1 away by positioning the cursor on the space after _'TSPUB at the bottom righthand corner of the screen, and typing ESC]U.

The CBSB at the top of the form means clear everything after the cursor position to the end of the screen, set background (i.e. display the normal white on black).

The 'TSPUB at the bottom right corner means return the cursor to the top lefthand corner of the screen (Ctrl T), set protect (i.e. move the cursor to the first underline after the cursor position and only allow subsequent cursor movement to occur where the underlines have been typed) and unset background (i.e. display any characters subsequently typed in reverse video if using a VT-100). On the LEX-11 VDU Form (see page 68), unset-background has been set to reverse video mode for the VT-100. There is no unset-background mode entered on the VDU Form for the VT52, since it has no useful property which could be switched on or off by this command, and hence the command will be ignored.

If this format does not suit the user's requirements, it would be advisable to alter this type of system record before the user enters many details, since if the length of the record were to increase, all existing entries would have to be brought onto the screen and written away again with ESC]U. This could be time-consuming if there were a lot of them.

THE DICTIONARY

Having established the input form for storing the names and addresses, one may immediately start entering them. However, before one can use them for lists, e.g. an alphabetical telephone list, it is necessary to set up a 'dictionary' record on the Memory File to define all the "fields" used, and this record like all other records must have a key of up to 10 characters (e.g. *DICT as shown below) followed by the information now described.

Each "field", i.e. string of characters, on the input form - such as Company Name, Address or Telex - must be given an identifying alphanumeric character and a name of no more than 8 characters (this name must contain no spaces but can contain a .) followed by the number of characters that have been allocated to that field. These field names and lengths must be set out from left to right, top to bottom, in exactly the order they appear on the input form. As an example, see the actual dictionary for *FORM1 set out below:

```
[*DICT
N NAME      36     A ADDRESS  144     P PHONE     12     T TELEX      8
1 CONTACT1  36     2 CLQNAME1  28     3 TITLE.1   36     4 CONTACT2  36
5 CLQNAME2  28     6 TITLE.2   36     7 CONTACT3  36     8 CLQNAME3  28
9 TITLE.3   36     I INSERT1   66     J INSERT2   66
L INSERT3   66
```

(N.B. There are two reserved fields, KEY and DATE, along with their identifying letters D and K, d and k, and these should therefore not be used)

When creating or updating a dictionary, position the cursor on the far left of the screen on the line below the last entry and type ESC]C or ESC]U as appropriate.

If there are several input forms (i.e. *FORM1, *FORM2 - each with their own record types), then each of them will require its own dictionary. The single character typed in front of each field name is called a formatting character and is used in formatting reports where columns or blocks of a fixed length are required rather than a variable length format - see the examples below. The field lengths above do not represent the amount of space occupied on the Memory File since strings of spaces are automatically squashed to a single character before writing to the Memory File. The records are similarly expanded on reading. The maximum length of field names in LEX-11 is 8 characters. While longer names may sometimes be accepted they may also cause run time errors. The format of the dictionary is free and is set out in columns only to assist readability. The up-to-8-character dictionary name must start with a letter and may contain upper or lower case letters, numbers or full stops.

The field name KEY is a reserved name and is always understood to be present. It represents the last 9 characters of the key of each record. Similarly, the field DATE is reserved. Today's date can be printed in a letter heading, report etc. by typing <DATE> where you wish it to appear, as in the letter heading below.

MASS MAILING

Once the input form for a particular style of name and address record has been established and entered into the Memory File along with its corresponding dictionary, LEX-11 can be used to selectively read the names and addresses for a variety of purposes, e.g. automatically addressing and typing the same letter to a selection of the names and addresses which have been stored (commonly known as mass mailing), addressing individual envelopes or printing sticky labels on continuous stationery, telephone lists, filling out standard forms or documents, addressing a one-off letter or for invoicing or other programmed applications.

The Mass Mailing Menu defines most of the Mass Mailing processing options set up in LEX-11 as released. The user can alter or add to these as he wishes. If the user sets up a new letter heading, new input form or any new facility, he must insert it into one of the menus by the means described in the previous

section on changing menus. In the following examples, the format *INDXL (a menu) allows the user to select which contact he wishes to write to, thus bringing into use the relevant format (say, *OUTL1) which sends a specified letter to that contact. These formats merely describe the format and content of the heading on the letter and you will be asked for the name of the document containing the body of the letter separately after having chosen option 1 or 2.

The letter is created in the normal way beforehand, using the standard Create option in the Master Menu. It should, of course, be constructed without the opening salutation as that is being picked up from the name and address record. It should also start with an embedded ruler - see example MAIL1.LTR below - since it may have certain information from the name and address records embedded in it, such as <INSERT1>. Since this information can vary in length widely from one name and address record to another, embedded rulers in the body of the letter are needed so that the letter printing routine can rejustify the text on its way to the printer.

```
[*INDXL     Mass Mailing Menu
L*OUTL1     1 - Letter to contact 1
L*OUTL2     2 - Letter to contact 2

            . - Select contact


[*OUTL1     *DICT P6066Y ?

<CONTACT1>, <TITLE.1>
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA                    <DATE>
 '


Dear <CLQNAME1>,

c
```

This mass mailing letter heading format may be set up in option 1 of the sytem menu or through a document file prior to writing it to the memory file. The lower case c at the bottom left corner is not actually typed - it is simply to show the position the cursor should be placed in before writing this record away to the memory file, using ESC]U. Following the key at the top left corner of the screen, the meaning of the successive groups is explained below:

```
[*OUTL1     *DICT P6066Y ?
```

Group	Meaning
[	Opening bracket required for any memory record
*OUTL1	The Memory File key of up to 10 characters

*DICT Commencing in column 12, the key of the dictionary which supplies the meaning of the field mapping characters, e.g. <CONTACT1><TITLE.1>,N,A in this case. It is terminated by a space as shown above.

P This says the output is to be Printed rather than Viewed.

V Replaces the P if the output is to be Viewed on the screen rather than Printed.

60 These two characters give the number of lines to be printed per page. (If the output were being viewed, the number of lines would be 21)

66 These two characters give the total number of lines on the page.

Y This space is for entry of a Yes (Y) or No (N) answer to the print question 'Pause at top of Form?'. Entry here of Y or N pre-answers the question, whereas if you are not sure whether your printer will have automatic paper-feed at the time you print your mass mailing then leaving the space blank means the question will be asked and you can decide at the time.

The next space after the Y or N is for entry of the record type, i.e. the first character of the key of the Name and Address records in this case (which is a blank). It enables LEX-11 to be sure of which of the many records on the Memory File contain name and address data. If you were to set up a record with a record type of say, a comma (,), then this position should contain , (a comma).

? The ? requests that LEX-11 offers the standard Selection option to the user at run time, i.e. Select_____
Part of the selection request may also be entered here, e.g. KEY=?, in which case you only need enter the key value, and not the 'KEY=', when you come to execute the selection request.

<CONTACT1> This requests the data contained in the CONTACT1 field (as shown in the dictionary) and could alternatively have been represented by <1> or a row of thirty six 1's.

<TITLE.1> This requests the data contained in the TITLE.1 field (as shown in the dictionary) and could alternatively have been represented by <3> or a row of thirty six 3's.

NNNNN etc The thirty six N's represent NAME as shown in the dictionary and could alternatively have been represented by <N> or <NAME>.

AAAAA etc The block of 144 A's represent ADDRESS as shown in the dictionary. If <ADDRESS> had been used, the entire address would have tried to print on one line and of course would generally not fit.

<DATE> This prints the date in conventional letter style format, e.g. 16th September, 1981

' The apostrophe indicates that anything following it should be printed as it appears and does not represent anything in the dictionary, i.e. Dear should be printed as such.

<CLQNAME1> This represents the colloquial name 1 as shown in the dictionary. It should always appear this way or as <2> since unused spaces in the field need to be ignored so that the comma appears directly behind the name.

MAIL1.LTR is printed below to give an example of a letter with insertions which can be brought in from the name and address records on the Memory File. The letter must start with an embedded ruler with further embedded rulers as appropriate:

```
.......F.............................................M.......................
      The problem of increasing software costs is
      widely acknowledged <CLQNAME1>.  Hardware costs
      are halving approximately every three years,
      salaries are rising by 10% per annum and,
      according to some pundits, programmer
      productivity is rising at best by 3% per annum.

.............J...............................M...................................
            <INSERT1>

.......F.............................................M.......................
      We would like to draw this to your attention
      <2>.  Please get in touch if you would like
      further information.

      Harry Jones
```

(Note that two spaces should be left after a paragraph with insertions in it).

In the last paragraph, the <2> is an alternative way of referring to Colloquial Name 1 since in the dictionary, 2 is shown as the formatting character for CLQNAME1. Note also that there is nothing in the above letter which tells LEX-11 which dictionary to use - it is the letter heading, *OUTL1 in this case, which gives the name of the dictionary (*DICT).

Address lines which are completely blank are not printed hence the space between the last line of the name and address and the Dear ... is constant. The line on which the <DATE> appears will always be printed and if this is the last line, say, of the address it may give the impression that the above number of blank lines is varying. If you wish to prevent this effect, put the <DATE> entry on the first or second line of the address, which presumably will always be printed, or place it on a line below the name and address.
The format (or system record) *OUTL2 below, addresses letters to the second contact, i.e. CONTACT2, and hence the formatting characters 1 and 3 for CONTACT1 and TITLE.1 become 4 and 6 for CONTACT2 and TITLE.2 respectively.

```
[*OUTL2    *DICT P6066Y ?

<CONTACT2>, <TITLE.2>
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA                    <DATE>
 '

Dear <CLQNAME2>,

c
```

(Again the c should not be typed, it merely indicates the position of the cursor immediately before writing the record to the Memory File by ESC]U).

If the letter which is to be produced with the above heading contains insertions of contact or colloquial name, as in MAIL1.LTR above, then these should of course be CONTACT2 and CLQNAME2. If the letters should be sent only to those addresses for which a CONTACT2 exists, which is likely, then the Selection should contain:

CONTACT2 NOT=vvv (vvv is shown here to mean spaces)

RECALLING PARTS OF MEMORY FILE RECORDS

Selected parts of Memory File records can be recalled onto the screen by specifying the displacement within the record and the length of text to be displayed. First, the cursor has to be positioned where you want the text to occur and then type:

[KEY+d/n ESC]R

The text of n characters starting from the character position d+1 in the Memory File record will be recalled subject to current ruler and mode (insert or overtype) settings.

STANDARD FORMS

The production of standard documents such as contracts or agreements can be batched so that they are all produced together. Such forms are like one of the letters described above except that there is no address at the top of it. Hence we need to define a letter heading with no formatting content so that we can use the standard facilities built in the Mass Mailing system. Such a letter heading without any formatting is shown below as the system record *AGREE :

```
[*AGREE    *DICT P6066N
c
```

(again the c indicates the position of the cursor prior to typing ESC]C or ESC]U)

All the information in this header, i.e. dictionary name etc., is used by LEX-11 when printing the standard form.

As an illustration of the kind of document we mean, the file AGREE.DOC could be created in option C of the master menu as follows:

EXAMPLE OF STANDARD DOCUMENT

<DATE>

```
F.............................................................M.............
This Document differs from an ordinary mass mailing letter in as
much as it does not have any Name and Address or salutation at
the beginning of it. A dummy heading *AGREE was used. Here we
insert CONTACT1:
<CONTACT1>
```

Typically it contains a number of inserts in different paragraphs. The same name may be inserted in several paragraphs, e.g. COMPANY NAME: <NAME> is inserted here for instance and INSERT1 here: <INSERT1>

```
.........F.............................................M......................
        Here is another paragraph with COMPANY NAME:
        <NAME> inserted once more followed again by
        INSERT1: <INSERT1> and PHONE:<PHONE>

F.............................................................M.............
```

Note that after a paragraph with one or more inserts in it, two blank lines must be left before the beginning of the next paragraph. This is the last paragraph.

LIST PROCESSING REPORTS

Reports are formatted in a very similar way to mass mailing headings and, in the same way, they have to be entered into a menu in order for the user to be able to select them. Reports use the S menu function.

Once again, the cardinal rule in setting up a report, until you become familiar with them, is to choose an existing report (i.e. take option 1 in the system menu and read the report record onto the screen) as a point from which to start and then change it according to your needs. Do not forget to also change the KEY to the new value before writing the format away.

In the three reports discussed below, the little c indicates the position of the cursor before writing the record to the Memory File by using ESC]U.

List processing reports are produced by either option S or option Q in the Mass Mailing menu, then selecting one of the further options, say option 2 from S, shown below:

S*OUTS2 2 - Name & Contacts
The report formatting record *OUTS2 lists Name & Contacts each in their own column. The same dictionary identifier (*DICT), Print/View Option (P), lines to be displayed and total length of screen or page (6066), the name and address record identification character, in this case a space, and request for Selection facility (?) apply as before, see below:

```
[*OUTS2     *DICT P6066          ?                          Contacts
 Key         Name                                          --------
 ---         ----

KKKKKKKKK    NNNN
```

wider than the screen width, e.g. to print a 132 wide report. If you need a continuation line but column 80 is required to be blank, then use the & character in column 80.

The first blank line after the heading lines is taken as a signal to LEX-11 that the heading has finished and the lines after the blank line are to be interpreted as formatting lines. Hence it is not possible to have a blank line in the heading, since the first blank line is interpreted as a switch and any further blank lines are assumed by LEX-11 to be part of the formatting lines so enabling double spacing etc. to be achieved. Even if one or more spaces is enclosed between apostrophes in an otherwise blank line in the heading, the line will still not print. You may use the 'Quoted blank', (default @), to obtain blank lines in headings, if necessary.

In the formatting line, the K's stand for the KEY of the record and, as the dictionary says, the N's and <CONTACT1><CLQNAME1> etc. stand for NAME, CONTACT1 AND CLQNAME1.

COLUMN TOTALLING IN THE LIST PROCESSOR

To total numeric columns in the list processing View or Print operations you indicate which field or fields you wish to total on the end of the *OUTQ/*OUTS line by means of /T:XYZ, where XYZ are the characters used in the *DICT to name the field/s. For example:

```
*OUTQ1  *DICT  V2023    ?    /T:O
Key          Company                                        Amount Owing
---          -------                                        ------------

KKKKKKKKKK NNNN
```

on the screen at a time. To make sure that part of the next record is not printed at the bottom of the screen and so progressively become a jumble of two records on the screen, simply make the display length equal to or less than the number of lines in the report including the heading. In this case, there are 21 lines per record and the display length could be set to 21 or less, which stops the next record from displaying but does not stop the initial record from being fully displayed. The actual names of fields have to be encompassed in apostrophes to distinguish them from the dictionary definitions, as follows:

```
[*ALLNA    *DICT V1623          ?
NAME AND ADDRESS RECORD LISTING IN FULL
---------------------------------------

'NAME KEY :' KKKKKKKKK
'NAME     :'NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
'ADDRESS  :'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
            AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
            AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 'PHONE:'PPPPPPPPPPPP
            AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA 'TELEX:'TTTTTTTT

'CONTACT 1):'1111111111111111111111111111111111111 2222222222222222222222222222
            3333333333333333333333333333333333333

'CONTACT 2):'4444444444444444444444444444444444444 5555555555555555555555555555
            6666666666666666666666666666666666666

'CONTACT 3):'7777777777777777777777777777777777777 8888888888888888888888888888
            9999999999999999999999999999999999999

'INSERT 1): 'IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
'INSERT 2): 'JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
'INSERT 3): 'LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
c
```

VARIABLE IN THE HEADING AND LITERALS IN THE FORMATTING LINES

The KEY, which of course varies from record to record, has been displayed in the heading of *ALLNA, by the Ks. The blank line following the heading still acts as a switch to say that what follows are formatting lines and, as before, this 'switch line' does not print.

In the lines which follow, any literals such as Name or Address must be enclosed in apostrophes. Remember the line must end before position 80.

The subsequent blank lines shown in the format, as before, will be printed. However, if a line containing formatting characters only (e.g. the second and third lines of A's in *ALLNA) has no data entered in it but only spaces, they will not print.

AUTOMATIC ADDRESSING OF A LETTER

This is an option in the Mass Mailing Menu (option C). When writing a one-off letter to someone whose name and address is already on the Memory File, it enables the particular name and address to be automatically printed at the top of the letter. It asks for the KEY of the name and address, as you will see if you choose the option, so that it can find the particular one required. (This

facility is probably unique to LEX-11 because it requires an indexed Memory File which few if any other word processors have). Note that we are only retrieving about half the information in the name and address record, e.g. we are not retrieving any of the INSERTS or the PHONE or the other CONTACTS.

The name and address could have been retrieved from all the information in the record at any point in the letter or document being constructed - this example extracts it at the beginning.

On choosing the Option, you will be asked for the key. Enter it and press Return. Next, you will be asked to enter the number given to the contact you wish to write to. After you have pressed Return, it will type the name and address and salutation for CONTACTn at the top of the screen.

The entry in the Mass Mailing Menu is shown below:

C*LETTR C - Create a new letter to existing contact

As you will see from the previous section on changing menus, this entry means Create a new blank file and then read the Memory File record whose KEY is *LETTR. This record consists mainly of programming with a small amount of text in it. Read Sections 5 and 6 on Programming if you are not familiar with programming. The contents of the record *LETTR are:

```
[*LETTR    $SQ Customer key\$SO!_'H\'U'V
2) < $0+201/23> 3) < $0+301/23> 4) < $0+401/23>
Which Contact -2, 3 or 4\$S9!_'H\
'T'V< $0+$901/36>
'V< $0+1/36>
'V< $0+37/36>
'V< $0+73/36>
< $0+109/36>
< $0+145/36>|{{{{{{{{{{{{{{{{{{{{{{{{{$Y

Dear < $0+$937/28>\'U$F   \,
```

We will comment on the above record line by line:

$SQ Customer key\$SO! 'H\'U'V

Unlike ordinary text files, standard paragraphs etc., all list processing records are space squashed so it is necessary to tell LEX-11 this before extracting information from a name and address record - hence the $SQ .

Next, the words 'Customer Key' are to be displayed on the screen.

It is necessary to use a keystore to store the key entered, hence $SO! . The $O! will stop the execution of the program and wait for the user to type the customer key, then press <return> to recommence recall of the program in the usual way. The 'H will then move the cursor left a space and \ will load the key into

keystore 0 and do a carriage return, leaving the cursor in the first column of the next line. 'U will move the cursor up a line and the 'V will delete the whole line, in particular it will remove the words 'Customer Key'.

2) < $0+201/36> 3) < $0+301/36> 4) < $0+401/36>
Which Contact -2, 3 or 4\$S9!_'H\

The expression bounded by the angle brackets reads 36 characters starting at the 201st character from a record whose key starts with a (space) character (i.e. it must be a name and address record because all of them start with a record type of space), and whose key continues with the contents of keystore 0 which have just been loaded by the user.

If you inspect the dictionary (i.e. the one whose key is *DICT in this case), you will see that CONTACT1 begins 200 characters from the beginning of the Name and Address record and it is 36 characters long. There is actually an extra character stored at the front of the record which requires us to add 1 onto the start position, hence the 201 in the program.

'T'V< $0+$901/36>
'T moves the cursor to the top of the screen and 'V clears to the end of the line, to clear the message previously displayed, then keystore 9 (contact number) for the record key held in keystore 0 with its location and length is displayed.

'V< $0+1/36> 'V to clear to end of line, then display Company Name.

'V< $0+37/36> 'V to clear to end of line, then display first line of address
'V< $0+73/36> 'V to clear to end of line, then display second line of address
< $0+109/36> Display third line of address
< $0+145/36>|{{{{{{{{{{{{{{{{{{{{{{{{{{{$Y
Display fourth line of address, then tab to M of ruler, backspace 23 spaces then recall today's date

Dear < $0+$937/28>\'U$F \,
The above line prints the 'Dear' and then reads the Colloquial Name corresponding to the chosen Contact (stored in keystore 9), returns to the beginning of the next line, moves up a line and finds the beginning of the first three spaces and puts in a comma.

Obviously, once you understand the above you can adapt the techniques to suit your particular name and address record or some other similar application.

SYSTEM MENU

The System Menu on the standard Memory File is shown below. Option 1 has already been described at the start of this Section. This illustrates that you can have the same item on as many menus as you please.

```
[*SYSTEM    System Menu. The following options are available:-
X*VSRN      0 - View memory record names
X           1 - Create/edit memory records
X*LISSYS    2 - Print system records

3           3 - Create a new Memory File
4           4 - Copy from one Memory File to another
X*VDUFORM   5 - Create/edit VDU details

S*PDTI      6 - Print document index
U)VT132AVO  7 - Change to 132 wide mode
T           8 - Memory File statistics

            . - SELECT OPTION
```

As you add and amend records on the Memory File, which is of a fixed length, it gradually fills up and it will be necessary to set up a new file as described below. If you wish to see how much space is left on the Memory File, choose the Statistics Option which will give the total size of the file in blocks (one block = 512 characters), the number used and the number still available. If you look at these statistics at the end of each editing session or at the end of each day, you will get a feeling for the rate at which space is being used up. If you are entering a lot of Names and Addresses, then space will be used up fairly rapidly but otherwise utilisation will be quite slow.

VIEW MEMORY RECORDS

This option will give you a list on the screen of the names of all the records on the Memory File.

PRINT SYSTEM RECORDS

This option prints out all records on the Memory File which begin with an *.

CREATING A NEW MEMORY FILE

LEX-11 writes system and other records into its Memory file (LEX11.VMF) and when only a given number (usually five) of data blocks are free LEX-11 will signal a warning message indicating that the file needs reorganising.

To reorganise the file it is necessary to first create a new memory file and then copy the the records across. This is achieved by selecting Option 3.

LEX-11 responds with the current memory file name and the statistics of the usage of that file :-

```
Stats of current file -   LEX11.VMF
No of data blocks used....    76
No of data blocks free....     4
No of keys used...........   145
No of keys on index.......   468
No of keys in overflow....     5
```

The number of data blocks used and free indicate current size of the file. The number of keys used is the number of records held on the memory file. The number of keys on the index indicates the total size of the index that was originally allocated and the number of keys in overflow, indicates that the number of keys that could not be accomodated within the original index size.

LEX-11 will then prompt for the new memory file name

New memory file name .....

This may be any valid file name (NEW.VMF) and is the file into which LEX-11 will reorganise the current file.

LEX-11 will then prompt

Manual / Auto Reorganise .

Enter M if you wish to control the reorganise or A if you want LEX-11 to allocate the parameters itself.

The next prompt is

New number of Data blocks ....

You enter the number of data blocks you wish to allocate to the new file, which will be based on the current size and the expected usage. Enter CR only to default to the current size.

The next prompt is

New no. of Keys on index ....

If auto is selected then LEX-11 will allocate an index size of twice the cuurent number of keys, subject to a minimum of 400. If manual, then you may enter any value, or CR only which will cause LEX-11 to take the same default as above.

LEX-11 then starts to create the file and indicates its progress with the message:-

Now creating file. Blocks written.. nnnn

If auto was selected LEX-11 will then copy all the records from the current file to the new one and will display the key of each record as it is copied. Otherwise you will control the copying as described in the section below.

When this is complete, the new memory file is ready , and you can rename or delete the old memory file and rename the file to LEX11.VMF.

COPY RECORDS FROM ONE MEMORY FILE TO ANOTHER

This facility is entered by selecting option 4 and allows you to copy records from your current memory file to any other memory file or vice versa.

LEX-11 will first prompt with the name of your current file and then asks for the other memory file name

Copy records. Current file - LEX11.VMF
Copy memory file name .....

and ask for the direction in which copying is to take place

Copy From or To current file F/T .

to which you respond F or T.

Key .....

is the next prompt which allows you to specify those keys which you wish to have copied. Note that all LEX-11 keys are ten characters in length with trailing spaces if necessary

Enter any existing key to cause that record to be copied.

Wildcards may also be used, so that where you enter a question mark in the key LEX-11 will copy all keys matching the non-wildcard characters.

i.e. ?????????? will copy all the records on the file
*????????? will copy all records commencing with *
*?DICT will copy all records with a first character of * and last eight characters of DICT .

Enter /+ after the key to copy all records with a key equal to or greater than the one specified.

Enter /Q after the key if you want LEX-11 to allow you to query each key. Note that this may be used in conjunction with /+.

When the query option is selected LEX-11 will display the each key, and you may respond with,

Y to copy the record
/ to rename the key. Here LEX-11 will prompt you for a new key name.
* to return to Key prompt.
Any other character will cause that record to be ignored and LEX-11 will proceed to the next key.

LEX-11 will always confirm when a record has been copied, and will make a check that the record does not already exist on the destination file. If it does then a warning message will be displayed and you may choose one of the options in the above table.

CHANGE MEMORY FILE WITHIN LEX-11

Menu Function Character F allows you to swap from one memory file to another from within LEX-11. You may use it in a Menu line followed by the name of the Memory file to swap to, or you can type '/F' and reply to the .VMF question with the name of the memory file to swap to. If the name is greater than 10 characters enter an invalid file name, which will cause another prompt and give enough space to enter the correct memory file name.

This feature is particularly useful under TSX+, since it allows users to have their own memory files on the same logical device. We suggest that under TSX+, that you set up a master LEX11.VMF which has **MENU containing F lines only, one for each user, to swap each user into his/her memory file.

SETTING UP DETAILS FOR DIFFERENT TERMINAL TYPES (VDUs)

You may set up LEX-11 so that it will operate with a variety of different makes and models of Video Terminals (VDU). Choosing option 5 in the system menu, or retrieving the system record *VDUFORM, by [*VDUFORM , gives you the following form to fill in:-

```
Details for [)_________ Type of Video Terminal
_=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___
_=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___

                                                Width ___   Lines ___
Set VDU modes ______________________________
Forward space _______   Insert line _______     Clear Screen _______
Up a line     _______   Delete line _______     Reverse LF   _______
Clear to EOL  _______   Insert char _______     BG clear EOS ___________
Clear to EOS  _______   Delete char _______     Cursor Addr  ___ _+___ _+___
Wraparound=BS _         IL does CR  _           Set FG ______ Set BG ______

ESCAPE Seq #1 0.......1.......2.......3....... ! # %&'()*+,-./0123456789:;<=>
____          ABCDEFGHIJKLMNOPQRSTUVWXYZ \ ^  abcdefghijklmnopqrstuvwxyz{|}

ESCAPE Seq #2 0.......1.......2.......3....... ! # %&'()*+,-./0123456789:;<=>
____          ABCDEFGHIJKLMNOPQRSTUVWXYZ \ ^  abcdefghijklmnopqrstuvwxyz{|}

Low value     0.......1.......2.......3.......
ASCII chars   ______________________________

              a _____ b _____ c _____ d _____ e _____ f _____ g _____ h _____
              i _____ j _____ k _____ l _____ m _____ n _____ o _____ p _____
              q _____ r _____ s _____ t _____ u _____ v _____ w _____ x _____
First Menu :- _________ Delay Factor _______ Write record _
```

This form is divided into four parts, which are now described as follows:-

VDUFORM PART 1

```
Details for [)_________ Type of Video Terminal
_=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___
_=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___ _=___
```

Here you enter the model no. of the VDU you are setting up, as the key, e.g. VT52, ADM3A , H1500 etc., followed by up to twenty local visible equivalents for the low value i.e non-visible ASCII characters. You enter the visible equivalent followed by the decimal value of the character. For example, to set $ to mean ESCAPE, and \ to mean CARRIAGE RETURN, set the first two fields to $=027 \=013

VDUFORM PART 2

```
                                                Width ___   Lines ___
Set VDU modes ______________________________
Forward space _______   Insert line _______     Clear Screen _______
Up a line     _______   Delete line _______     Reverse LF   _______
Clear to EOL  _______   Insert char _______     BG clear EOS ___________
Clear to EOS  _______   Delete char _______     Cursor Addr  ___ _+___ _+___
Wraparound=BS _         IL does CR  _           Set FG ______ Set BG ______
```

This is where you enter the character strings which must be sent to the VDU to cause it to carry out each of the operations indicated. Where a non-visible character, e.g. ESCAPE, forms part of a string, then enter it in the visible equivalent table above, and enter its visible equivalent here. If the VDU does not have the hardware feature available, then leave the field blank and LEX-11 will execute the function by software.

FIELD	ENTER
Width	The number of characters available on one line
Lines	The number of lines available on the screen
Set VDU modes	Any character sequence you may wish to send to the VDU at LEX-11 start-up time, for example to set the VDU into a particular mode.
Forward space	The character sequence which when sent to the VDU causes the cursor to move one space forward
Up a line	The character sequence which when sent to the VDU causes the cursor to move up one line
Clear to EOL	The character sequence which when sent to the VDU clears data from the current cursor position to the end of the line.
Clear to EOS	The character sequence which when sent to the VDU clears data from the current cursor position to the end of the screen.
Insert line	The character sequence which when sent to the VDU causes a new empty line of blanks to be inserted at the current cursor position, and all subsequent lines on the screen to be moved down by one line. If the action of this character string also causes the cursor to end up at the beginning of the new line inserted, rather than staying where it was at the time of the insert line, then enter Y in the 'IL does CR' field below.
Delete line	The character sequence which when sent to the VDU causes the current line to be deleted and all subsequent lines on the screen to be moved up one line.
Insert char	The character sequence which when sent to the VDU causes all characters on the current line to the right of the cursor to be moved right one position, leaving a blank character at the current cursor position.
Delete char	The character sequence which when sent to the VDU causes all characters on the current line to the right of the cursor to be moved left one position.
Clear Screen	The character sequence which when sent to the VDU causes all data on the screen to be cleared
Reverse LF	The character sequence which when sent to the VDU causes reverse scroll to occur if the cursor is positioned on the top line of the screen. If the VDU has no specific reverse line feed command, but does have insert line, then enter the insert line sequence here.
BG clear EOS	The character sequence which when sent to the VDU causes the whole screen to be cleared to background spaces
Cursor Addr	The character sequence which when sent to the VDU causes the next two characters to be treated as coordinates to position the cursor. This is then followed by an X or a Y in the next column to indicate whether the row (X) or column (Y) is the first coordinate of the two coordinate characters.

You follow this by the number that needs to be added to the absolute row or column number, before sending it to the VDU.

The next pair of fields _+___ , is where you enter a code for the type of cursor positioning used on the VDU, followed by the amount to be added to the second coordinate, i.e row if the first coordinate was column, or column if the first coordinate was row.

The valid cursor positioning type codes are:-

V - Straightforward coordinate + bias, e.g VT52, ADM3A

A - ANSI mode coordinates, e.g VT100

H - Hazeltine 1500 discontinuous bias, for correct operation enter X+096 H+032

Wraparound=BS Enter the visible equivalent of Back space here if the VDU cursor wraps round from the last position on a line to the first position on the next line whenever a character is displayed in the last position on a line.

IL does CR Enter 'Y' here if you have entered anything in the Insert Line field, and the insert line command causes the cursor to position itself at the start of the line when it is finished.

Set FG Enter the character sequence which when sent to the VDU causes characters to be displayed in full intensity.

Set BG Enter the character sequence which when sent to the VDU causes characters to be displayed in half intensity.

VDUFORM PART 3

```
ESCAPE Seq  1  0.......1.......2.......3....... ! # %&'()*+,-./0123456789:;<=>
____           ABCDEFGHIJKLMNOPQRSTUVWXYZ \ ^  abcdefghijklmnopqrstuvwxyz{|}

ESCAPE Seq  2  0.......1.......2.......3....... ! # %&'()*+,-./0123456789:;<=>
____           ABCDEFGHIJKLMNOPQRSTUVWXYZ \ ^  abcdefghijklmnopqrstuvwxyz{|}

Low value      0.......1.......2.......3.......
ASCII chars    ______________________________
```

This section of the form is where you translate the special (Non-QWERTY) keys on the keyboard of your VDU into the internal operations that LEX-11 can perform. Each special key on a VDU keyboard can generate either a low value ASCII character (Value below space, ie decimal 32), or a sequence of two or more characters known as an ESCAPE Sequence. An ESCAPE Sequence is so called because the first character of the sequence is usually the ASCII ESCAPE character (Value Decimal 27).

The internal operations that LEX-11 can perform are identified by an upper case letter, from the table given below. You can use a lower case letter to define a combination of operations or a LEX-11 program function to be performed for a single key depression. In this case you enter the desired combined operation against the corresponding lower case letter in the last part of this screen.

INTERNAL LEX-11 OPERATION CODES

01	A	Insert Line	13	M	Carriage Return
02	B	Back word	14	N	Insert Character(Use ruler)
03	C	Spare	15	O	Spare
04	D	Delete Character	16	P	Delete Character(Use ruler)
05	E	Insert Character	18	R	Cursor Right
06	F	Forward word	20	T	Cursor to Top of screen
07	G	Get to end of text on line	21	U	Cursor Up
08	H	Cursor left (Back Space)	22	V	Clear to end of line
09	I	Horizontal Tab	23	W	Delete word
10	J	Cursor down (Line Feed)	24	X	Delete line
11	K	Change case of current word	25	Y	Duplicate line above
12	L	Down 23 Lines (Form Feed)	26	Z	Spare

There are three tables where you enter the LEX-11 operation to be performed by each special key. Two are tables for ESCAPE Sequences, and one is for the low value ASCII characters. For the ESCAPE Sequence tables, you enter in the first four character field the sequence of characters that introduces the escape sequence. This is usually one or two characters, the first of which is normally ESCAPE. You then use the 128-character table to translate the character which identifies the key into the internal LEX-11 operation. You do this by entering the single character code for the LEX-11 operation underneath the value of the character which identifies the key.

In the case of a VT52, the arrow keys send the escape sequence ESC A, ESC B, ESC C, ESC D for Up, Down, Right, and Left. In this case the sequence which introduces the escape sequence is simply ESCAPE, and the characters which identify the key for the arrow keys are A, B, C, D. So underneath the position corresponding to ABCD in the 128-character table, you enter UJRH, which are the internal LEX-11 codes for Up, Down, Right, and Left.

For a key that generates a low value ASCII character on its own, you enter the desired internal LEX-11 operation code in the corresponding position for that key in the Low Value ASCII table.

If you want to define a combination of operations to be performed for a single keystroke, enter a lower case letter between 'a' and 'x' underneath the character corresponding to the key in the translate table. Then use Part 4 of the screen, shown below, to define the combination of operations you require to be executed for that key. Against the lower case letter chosen, enter the sequence of characters required. If this sequence is longer than the five spaces allowed for each letter, you may use the space against the next letter to continue, but this stops you from using the next letter for another combined operation.

Note that in Part 4 you must use visible equivalents of the low value ASCII characters if you wish these to form part of the combined operation.

VDUFORM PART 4

a _____	b _____	c _____	d _____	e _____	f _____	g _____	h _____
i _____	j _____	k _____	l _____	m _____	n _____	o _____	p _____
q _____	r _____	s _____	t _____	u _____	v _____	w _____	x _____

The example opposite shows how the numeric keypad on a VT52 can be used to carry out some of the cut and paste operations using single keys. The VT52 in numeric keypad mode generates ESCAPE Question Mark followed by the lower case letters p-y for the numbers 0-9.

THE **TEXT SYSTEM RECORD

The system memory file record **TEXT contains most of the general LEX-11 system configuration details. It contains -

1. Error and informational message texts
2. Main menu heading
3. Printer definitions
4. Operating System dependent information
5. Meta Character definition
6. Default System device and VDU form

**TEXT can be edited under Option 1 of the system file menu using the ESC]R and ESC]U functions. Once edited and replaced, its effect will only be felt by exiting LEX-11 and re-entering. The configuration details are examined on start-up only.

Below is an example of a distributed **TEXT record. It consists of a sequence of text and number fields delimited by a \ or end-of-line. Most of the fields are straight message text fields and should be self-explanatory. You are at liberty to change any of the messages that you might like. The lines marked * contain system configuration options and are examined and explained in detail below.

Example **TEXT Record

```
[**TEXT    License No. ---- For ?????? ** NOT FOR RESALE **
.BAK\.LTR\$~^                                                    *
Which Printer (1) Slow  (2) LP: (3) Spool to a file !            *
TT1:  A\LP:\ Spool File Name ! \\\\                              *
5\WARNING. Memory File is nearly full^\Type of V D U = \VT1ZZ\S\SY:  *
Document Title:\   Old\   New\ Input\Output\Document Name : \Selected. WAIT ..*
 already exists^\ doesn't exist^\ is not a valid *DICT entry^
LEX-11 is searching\Using Format :- \Can't find that format^
Answer HELP to next question for further assistance
Select:\HELP\Bad selection^\Press any key to continue, * to exit^\ \
Pause at top of form !\Line up paper, then press any key to continue^
\\\Missing Menu
Searching for: \No *DICT\Memory File Statistics
Total Blocks \Blocks used  \Blocks free  \FINISHED. Press any key to continue^
\Key\\Number of records selected:\These field names may be used to select
String: \ Not found
AND     &&OR      /&NOT     @@KEY     KSRECTYPE kSDATE    DS TOTAL...
 \Start Page No.  \Skip till page  \End page number
Invalid ruler in document format\Number of Copies !
```

**TEXT Configuration Fields

Each asterisked line above is detailed below with notes on the important fields.

1. [**TEXT Licence No. ---- For ?????? ** NOT FOR RESALE **

Main Menu title

2. .BAK\.LTR\$~^

.BAK is the backup file type ,
.LTR is the default document type
$~^ are printer control characters ($=ESC,~=no LF,^=control)

3. Which Printer (1) Slow (2) LP: (3) Spool to a file !

Printer Option questions and codes

4. TT1: A\LP:\ Spool File Name ! \\\\

Printer Device names/addresses - TT1: A = printer 1 (daisywheel)
- LP: = printer 2 (line printer)
- Spool = option 3 (spool file question)

Program mode - If set to A, then the initial program mode will be charcter interpretation , otherwise it will be functions.

5. 5\WARNING. Memory File is nearly full^\Type of V D U = \VT1ZZ\S\SY:

No of Blocks in Memory File at which warning message will appear
Video terminal change message
VDU Prompt
VDU Form name
Operating system- V-VAX S-otherwise
Logical Device to write documents to

6. Document Title:\ Old\ New\ Input\Output\Document Name :

Enables document titling (can be removed to disable titling)

INSTALLATION NOTES

LEX-11 runs on a number of different PDP-11 and VAX operating systems. At present it is supported under RT11, TSX+, RSTS, RSX11M, and VAX/VMS. IAS and UNIX versions are under development. Each operating system provides a different environment for running LEX-11. The following notes give installation procedure details and hints for successful use of LEX-11 on each particular operating system.

RSTS

The following notes are intended to be used by a RSTS Systems Programmer operating in a privileged account to enable the installation of LEX-11.

1. Load files from supplied mag tape or floppy disc.

For mag tape: PIP *.*=MT0:[*,*]*.*/BL

For floppy RUN $CTSPIP

SY:=DX:*.*

The files copied will be:

LEX11.S11	the LEX-11 word processor program
LEX11.VMF	the memory file
LEX11.RTS	the LEX-11 run-time system

2. Load and Add the LEX-11 Run-time System

LEX-11 runs under the control of a run-time system that must reside in [0,1] and be ADDed as a run-time system.

```
PIP (0,1)/MO:16=LEX11.RTS

UT ADD LEX11
```

3. Make the LEX11 program runnable. :

```
PIP  LEX11.S11<104>/RE

UT NAME LEX11=LEX11.S11
```

Notes

If LEX11 is being installed permenantly, the UT ADD LEX11 should be added to the start-up control file.

LEX-11 may be invoked by a CCL, which can be added by:

```
UT CCL LEX-=[1,2]LEX11.S11
```

If this CCL is required permenantly, then it also must be added to the start-up control file.

LEX-11 requires that each account that is to use it must have its own copy of the memory file - LEX11.VMF. This file, among other things, contains the user's document directory and standard paragraphs, etc. If write access is required to the VMF, then each user should have their own copy of LEX11.VMF. If document titling is enabled, it implies write access.

The default device on which documents are created for each user is specified in a field in the **TEXT system record. See Section 16).

The run-time system is 7K words. Each copy of the LEX-11 program running is 16K words. LEX-11 uses binary keyboard mode extensively.

RT11 and TSX+

The RT-11 version generally comes on RX01 floppy disk media, although it can come on other media by arrangement. The standard distribution disk comes as an unbootable disk with 2 files on it. The files are:-

LEX11.SAV	the LEX-11 program
LEX11.VMF	the memory file

The program, LEX11.SAV, must reside on logical unit SY:. The memory file LEX11.VMF is normally on logical unit DK: but it is possible to have it on another device by using appropriate start-up procedures (possibly in command files). The document files are created on the logical unit defined in the **TEXT system record. See **TEXT section at the back of this chapter.

The device handlers for all devices to be accessed must be LOADED before LEX-11 is started. Printers must be operable by standard PIP functions through device handlers, in order for LEX-11 to access them. Printer unit names are specified to LEX-11 in the **TEXT system record.

```
LOA TT
LOA LP          (LA-120 for example)
LOA LQ          (Qume   for example)
SET USR NOSWAP
```

To start LEX-11, you specify the LEX-11 program name.

```
R LEX11
```

When LEX-11 is exited by the user, it returns to the monitor.

RSX11M

LEX-11 runs on version 3.1 or upwards of RSX11M, and requires a PDP-11 with memory management and EIS. The SYSGEN for RSX11M must include these options:

Mapped
Files-11 ACP
Get task parameters, extend task
Checkpointing with system checkpoint files
Unsolicited input AST

In addition it is advisable that the full duplex terminal driver be included, for improved character typeahead handling.

LEX-11 is delivered on either a DOS format tape or an RT11 format RX01, or other media by arrangement. Use FILEX to transfer the files, as follows:

For Mag Tape: FLX SY:/RS/IM=MT:[*,*]*.*/DO

For Floppy Disc: FLX SY:/RS/IM=DX:*.*/RT

The files transferred, and their approximate sizes, in disc blocks should be:

LEXRTS.TSK	33	) these files must
LEX.TSK	9	) be contiguous
LEX11.L11	192	
LEX11.VMF	84	

The LEX-11 run-time system is implemented as a shared library, which requires a dedicated COMMON partition of 8K words, called LEXRTS, to be allocated.

The following procedure may be used to allocate the partition and install LEX.

```
SET /UIC=[1,54]
RUN VMR
RSX11M
SET /TOP=GEN:-400              ** Ignore Warning Message **
SET /MAIN=LEXRTS:*:400:COM
INS LEXRTS
INS LEX
^Z
reboot
```

This takes 8K away from the top of GEN and gives it to LEXRTS. LEX is invoked by typing LEX at the users terminal. If there is a LEX11.VMF in the users account, it is used as the .VMF, otherwise it looks for LX:LEX11.VMF in the user's account, and uses that if it can find it. Otherwise it looks for LB:[1,54]LEX11.VMF and uses that if it can find it. The same rules about sharing .VMF files apply as with RSTS.

VAX-VMS

These notes apply to version 2, i.e. the shared version of the LEX-11 run time system. The following files are required:

File	Approx size
LEX.EXE	7
LEXRTS.EXE	41
LX11.L11	192
LEX11.VMF	84

The files are distributed on either DOS format Mag Tape, or RT11 format Floppy Disc. They may be transferred using FILEX as follows:

For Mag Tape:

```
MOUNT MTA0:/FOREIGN
MCR FLX
SY:/RS/IM=MT0:[*,*]*.*/DO
^Z
```

For Floppy:

```
MCR FLX
SY:/RS/IM=CS1:*.*/RT
^Z
```

FLX may say 'Illegal Device' to the CS1:, in which case you may need to enable the floppy by means of:

```
RUN SYS$SYSTEM:SYSGEN
CONNECT CONSOLE
^Z
```

You must copy the run-time system into SYS$SHARE, by means of:

```
COPY   LEXRTS.EXE  SYS$SHARE:LEXRTS.EXE
```

and LX11.L11 into SYS$LIBRARY by:

```
COPY   LX11.L11    SYS$LIBRARY:LX11.L11
```

To run LEX-11, you type RUN LEX. You may wish to set this up in a command file to run LEX from a common account.

You must copy the run-time system into SYS$SHARE, by means of:

```
COPY   LEXRTS.EXE  SYS$SHARE:LEXRTS.EXE
```

and LX11.L11 into SYS$LIBRARY by:

```
COPY   LX11.L11    SYS$LIBRARY:LX11.L11
```

To run LEX-11, you type RUN LEX. You may wish to set this up in a command file to run LEX from a common account.

Function	Control keys	Visible Equivalent
Pressing the control key together with another character enters one of the ASCII control characters.	Ctrl	'
Pressing the escape key indicates that the following character(s) have a special meaning. These short strings are called 'escape sequences'.	ESC	$
Move cursor UP a line	Up arrow	'U
Move cursor DOWN a line	Down arrow or linefeed	'J
Move cursor RIGHT one space	Right arrow	'R or }
Move cursor LEFT one space	Left arrow	'H or {
DELETE last character typed	Delete or rubout key	
Cursor to next TAB position	Tab	'I or |
Cursor to start of NEXT LINE	Return	'M or \
CHANGE CASE from cursor position to end of word	Ctrl and K	'K
Move to END OF typing LINE	Ctrl and G	'G
Move FORWARD a word	Ctrl and F	'F
Move BACK a word	Ctrl and B	'B
INSERT A SPACE in existing text (say, between two characters)	Ctrl and N (PF3 on VT-100) (where margin or tab stops should be observed) (use Ctrl and E where margin and tab stops should be overridden)	'N 'E
DELETE A SPACE in existing text	Ctrl and P (PF4 on VT-100) (where margin or tab stops should be observed) (use Ctrl and D where margin and tab stops should be overriden)	'P 'D
INSERT a blank LINE (inserts blank line at cursor position, moving all subsequent text down by one line)	Ctrl and A (PF1 on VT-100)	'A
DELETE a LINE (deletes line where cursor is positioned moving all subsequent text up one line)	Ctrl and X Ctrl and B (under RSX op.sys) (PF2 on VT-100)	'X
'UNDELETE' a LINE	ESC. Ctrl and X	

Function	Control keys	Visible Equivalent
DELETE REMAINDER of line (deletes from cursor to end of line)	Ctrl and V	'V
DELETE WORD	Ctrl and W	'W
DISPLAY 1 screenful of text (displays screenful from cursor position - equivalent to 23 line feeds)	Ctrl and L	'L
TOP of screen (moves cursor to top of screen, then use Up arrow to move further back or ESCV to return to beginning of the document)	Ctrl and T	'T
COPY from line above (copies character-by-character from line directly above)	Ctrl and Y	'Y
INSERT TEXT (put cursor at point where text is to be added. As typing begins, text to right of cursor is pushed along and onto new lines)	ESC followed by I	$I
REJUSTIFY TEXT (and also exit insert mode if in use). Tidies up a paragraph after insert mode or, together with a J.....M. ruler, redistributes text to a straight righthand margin	ESC followed by J	$J
EXIT INSERT MODE (does not rejustify paragraph)	ESC followed by -I	$-I
BREAK text for NEW LINE (moves text following cursor position to new line. Insert line for a new paragraph)	ESC followed by LineFeed	$'J
CENTER TEXT (centers text between two tab stops without using a C.....M. ruler)	ESC followed by @	$@
CURRENT DATE in full	ESC followed by Y	$Y
CURRENT DATE in short	ESC followed by T	$T

Function	Control keys	Visible Equivalent
FIND text (type ESC F followed by word or phrase you wish to find, press return and cursor moves to the specified text)	ESC followed by F	$F
GET text (alternative method to find a string in the text. Type ESC G followed by word or phrase you wish to find, press return and cursor scrolls through the document to the specified text)	ESC followed by G	$G
HELP messages	ESC followed by H	$H
NEXT RECORD - obtain next highest key in the memory file	ESC followed by N	$N
KEYSTORE setting	ESC followed by Sn	$Sn (n = number 0 - 9)
KEYSTORE recall	ESC followed by n	$n
VERTICAL character entry	ESC followed by \	$\
EXIT VERTICAL character entry	ESC followed by \	$\
Cut and paste functions		
MARK TEXT	ESC followed by (	$(
CUT AND LEAVE TEXT	ESC followed by)L	$)L
CUT AND BLANK TEXT	ESC followed by)B	$)B
CUT AND SQUEEZE TEXT	ESC followed by)S	$)S
PASTE AND OVERLAY TEXT	ESC followed by *O	$*O
PASTE AND INSERT TEXT	ESC followed by *I	$*I
PASTE AND 'ELBOW' TEXT	ESC followed by *E	$*E
MERGE OPEN - merge a document into document currently being edited	ESC followed by MO	$MO
MERGE ADVANCE - advance to string in document to be merged	ESC followed by MA	$MA
MERGE UNTIL - copies lines from merge document until string is found	ESC followed by MU	$MU

Function	Control keys	Visible Equivalent
MERGE TO END - merges any remaining lines from the merge document into the current document	ESC followed by ME	$ME
MERGE INSERT - inserts one line from the merge document into the current document	ESC followed by MI	$MI
REPEAT function	ESC followed by X	$X
(type ESCX, no. of repeats and function to be executed and finish with Return, e.g. ESCX6. would give ..... To repeat indefinitely define no. of repeats as zero, e.g. ESCX0'J would repeat Line Feed until the end of input file)		
Enter CALCULATOR	ESC followed by +	$+
BEGINNING of document (re-edit) (exits, writes document away on disk and re-enters at beginning for further editing)	ESC followed by V	$V
EXIT EDIT or move back to Main Menu	ESC followed by E	$E
QUIT EDIT (exits without updating document)	ESC followed by Z	$Z
PRINT-time command to extend lines beyond 80 characters to width set on printer	ESC followed by &	$&
RULER control		(n = ruler no.0 - 9)
Display ruler no.?	ESC followed by Mn	$Mn
Use (no display) ruler no.?	ESC followed by Un	$Un
Write ruler as no.?	ESC followed by Wn	$Wn
SYSTEM record handling		
Enter a record name	[KKKKKKKKK	[
Create a new record	ESC followed by]C	$]C
Update an existing record	ESC followed by]U	$]U
Read an existing record	ESC followed by]R	$]R
Use system record and translate any visible equivalents	]	]

Function	Control keys	Visible Equivalent
Store system record without carriage returns (as with Standard Paragraphs):		
Create a new record	ESC followed by >C	$>C
Update an existing record	ESC followed by >U	$>U
Delete an existing system record	ESC]K (after key)	$]K
Instruction to LEX-11 to translate text stored within double quotes literally upon recall or withhold execution of escape sequence until later recall	" "	
Instruction to LEX-11 to translate next character literally upon recall rather than as a visible equivalent to be executed, e.g. a ! stored within a standard paragraph will be returned as a pause point rather than an exclamation mark unless preceded by @ (i.e. No way@!)	@	@
Set background - set the screen into 'background' mode. The actual mode used for any particular terminal can be defined by the user, eg VT100 can be set into its normal mode (white on black)	ESC followed by SB	$SB
Unset background - set the screen into a 'foreground' mode. The actual screen mode used on a particular terminal can be defined by the user, eg set VT100 into reverse, i.e. black on white	ESC followed by UB	$UB
Clear background - clear screen from current cursor position to end. When a terminal with foreground/background feature is being used, clear background only.	ESC followed by CB	$CB
Clear foreground - clear the screen from cursor position to end. If the terminal has foreground/background feature, clear foreground only.	ESC followed by CF	$CF

Function	Control keys	Visible Equivalent
Set protect - the Set Protect Mode restricts the cursor to 'unprotected' areas of the screen which have previously been indicated by strings of underline characters	ESC followed by SP	$SP
Unset protect - Unset the Protect Mode restriction set by the ESCSP command	ESC followed by UP	$UP
Clear protect - Unsets protection set by the ESCSP command and clears them to blanks. Also clears unprotected underlines to blanks	ESC followed by CP	$CP
Shush Mode - disable display on the screen of characters entered. The cursor moves but there is no display	ESC followed by SH	$SH
Squash Character - squash out repeated character, i.e. ESCSQc would define c as the current 'squash character' and multiple occurrences of c in the system record would be stored in a single byte	ESC followed by SQ	$SQ
Open Document - create a document when in option 1 of the system file menu. Type ESCOD followed by document name and Return	ESC followed by OD	$OD
Send to Video - sends string to screen without putting into screen buffer. Type ESCSV followed by string which should start and end with a delimiter, e.g. ESCSV.string.	ESC followed by SV	$SV
PAUSE point for entry of data when a system file is recalled in use (i.e. with])	Embed ! within a system file record (such as a standard paragraph) at point where entry of variable text is required	!

Function	Control keys	Visible Equivalent
Typed Instructions to be executed at Print Time		
Instruction to move to next sheet or pause in print	.P (type this in document at the very left of screen at point where new page is required)	
TITLING, automatic reproduction of a title on each new page at print-time	.Tn,n (type this in document at the very left of screen at the top of the document)	
FOOTERS, automatic reproduction of a footer on each new page at print-time	.F (type this in document somewhere through first screen - two lines are reserved for this, so even if only one is required leave a blank line)	
PRINT TIME SPECIAL EFFECTS	.C Followed by the print-time visible equivalents for special effects, in the order:	
	1. ESCAPE	$
	2. Suppress Line Feed	~
	3. CNTRL	^
	4. ESCAPE SPACE (pounds)	#
	5. ESCAPE \ (double underline)	=
	6. SPACE	@
	7. Paragraph Numbering	!
	8. Indexing	|
VARIABLE LINE SPACING	.S n - where n equals one of the following: .5, 1, 1.5, 2, 2.5 or 3	
INFORMATION	.I followed by information e.g. to reset paragraph number element	